Petit guide de mise en place d'un mandataire transparent avec Linux et Squid

Version française du *Transparent Proxy with Linux and Squid mini-HOWTO*

Daniel Kiracofe

Traduction française: Geneviève Gracian

<ggracian CHEZ free POINT fr>

Relecture de la version française: Jean-Philippe Guérard

<jean TIRET philippe POINT guerard CHEZ corbeaunoir POINT org>

Version 1.15.fr.1.0

14 juillet 2003

Historique des versions

Version 1.15.fr.1.0	14 juillet 2003	
	Mise à jour de la version française.	
Version 1.15	août 2002	
Version 1.13.fr.1.0	19 janvier 2003	
	Première version française.	
Version 1.13	janvier 2002	

Ce document détaille pas à pas la mise en place d'un serveur mandataire transparent, en n'utilisant que Linux et Squid. Il traite aussi bien de la configuration du noyau, de la configuration des règles iptables, de la configuration réseau, que de la configuration du serveur mandataire lui-même.

Table des matières

1. Introduction

1.1. Commentaires

Les commentaires et réactions générales à propos de ce petit guide sont les bienvenus et peuvent être adressés en anglais à son auteur Daniel Kiracofe `<drk CHEZ unxsoft POINT com>`.

Les commentaires, corrections et suggestions relatifs à la version française de ce document, et aux liens qu'elle contient sont les bienvenus et peuvent être adressés à `<commentaires CHEZ traduc POINT org>`.

1.2. Droits d'auteur et marques déposées

1.3. #include <dénégation.h>

Aucune garantie explicite ou implicite, et cætera, et cætera, et cætera.

2. Vue d'ensemble de l'utilisation d'un mandataire transparent

2.1. Motivation

Lors de l'utilisation d'un mandataire (proxy) « ordinaire », le client indique à son navigateur le nom d'hôte et le numéro de port du serveur mandataire. Le navigateur dirige alors ses requêtes vers le serveur mandataire qui les redirige vers les serveurs cibles. Cependant, de temps en temps, on se trouve dans l'une des situations suivantes. Soit :

- vous voulez obliger les clients de votre réseau à utiliser le serveur mandataire, qu'ils le veuillent ou non ;
- vous voulez que les clients utilisent le mandataire mais vous ne voulez pas qu'ils le sachent ;
- vous voulez que les clients passent par le serveur mandataire, mais vous ne voulez pas faire tout le travail nécessaire à la mise à jour des réglages de centaines ou de milliers de navigateurs.

C'est ici que le mandataire transparent entre en scène. Une requête *web* peut être interceptée de façon transparente par le mandataire. Pour autant que le sache le client, il est en train de parler au serveur d'origine, alors qu'il communique en réalité avec le mandataire. (Notez que la transparence ne s'applique qu'au client ; le serveur sait qu'un serveur mandataire est mis en œuvre et voit son adresse IP, et non celle de l'utilisateur. De plus, Squid a la possibilité de transmettre un en-tête `X-Forwarder-For` au serveur, afin que celui-ci puisse déterminer l'adresse IP réelle du client).

Les routeurs Cisco peuvent être utilisés comme mandataires transparents ainsi que de multiples commutateurs. D'une manière assez épatante, Linux peut être configuré comme routeur, et servir de mandataire transparent en redirigeant les connexions TCP vers des ports locaux. Cependant, il est nécessaire de s'assurer que le serveur mandataire soit au courant de cette redirection, afin qu'il puisse se connecter aux véritables serveurs de destination. Il existe deux moyens généraux pour cela :

- Tout d'abord, lorsque le serveur mandataire n'est pas capable d'agir en tant que mandataire transparent, vous pouvez utiliser un petit démon astucieux appelé Transproxy qui siège devant le mandataire et s'occupe de tous les détails triviaux à votre place. Transproxy a été écrit par John Saunders, et peut être trouvé sur http://www.transproxy.nlc.net.au/. Transproxy ne sera pas présenté plus en détail dans ce document.
- Une solution plus propre est d'utiliser un serveur mandataire directement capable d'agir en tant que mandataire transparent. Celui sur lequel nous allons nous pencher est Squid. Squid est un serveur mandataire pour Unix, dont les sources sont publiques, et qui est capable de mémoriser les pages *web*. On peut le trouver sur http://www.squid-cache.org.

Il est également possible, au lieu de rediriger les connexions vers des ports locaux, de les rediriger vers des ports distants. Ceci sera traité dans Section 6, « Mandataire transparent pour une machine distante ». Les lecteurs intéressés par ce sujet devraient aller directement à cette section. Les lecteurs qui souhaitent mettre en place sur une même machine la redirection et le mandataire peuvent faire l'impasse sur cette section.

2.2. Étendue du document

Ce document se concentre sur la version 2.4 de Squid ainsi que sur la version 2.4 du noyau. Ces versions sont les plus récentes versions stables au moment de son écriture (août 2002). Il devrait également s'appliquer à la plupart des noyaux 2.3 les plus récents. Si vous souhaitez utiliser des versions antérieures de Squid ou de Linux, vous pouvez vous référer à http://users.gurulink.com/drk/transproxy/. Notez que ce site a déménagé.

Si vous utilisez une version de développement du noyau ou de Squid, vous serez livré à vous-même. Ce document peut vous aider mais c'est vous qui voyez.

Notez que ce document ne traitera que des mandataires HTTP. Je reçois une foule de messages au sujet de la mise en place de mandataires FTP transparents. Squid ne possède pas cette capacité. Il semblerait qu'un programme nommé Frox le puisse. Je ne l'ai pas essayé, donc j'ignore s'il fonctionne bien. Vous pourrez le trouver sur http://frox.sourceforge.net/.

Ce document se consacre essentiellement à Squid. Cependant, Apache peut aussi être utilisé comme mandataire avec mémoire des pages. (Si vous hésitez sur le serveur mandataire à adopter, je vous recommande Squid. En effet, Squid a été pensé dès le départ comme serveur mandataire. Apache, de son côté, ne s'est vu rajouter les fonctionnalités de mandataire qu'après coup). Si vous désirez utiliser Apache au lieu de Squid, suivez toutes les instructions de ce document relatives au noyau et aux règles iptables. Ne tenez pas comptes des sections spécifiques à Squid, et allez voir les sources et le mode d'emploi du module mandataire transparent pour Apache sur http://lupo.campus.uniroma2.it/progetti/mod_tproxy/. (Merci à Cristiano Paris `<c POINT paris CHEZ libero POINT it>` pour sa contribution sur ce point).

2.3. HTTPS

Enfin, en ce qui concerne la mise en place d'un mandataire transparent pour HTTPS (par exemple, pour les pages *web* utilisant SSL, TSL, et cætera), *vous ne pouvez pas le faire*. Ne le demandez même pas. Pour comprendre pourquoi, effectuez une recherche avec les mots clefs « attaque de l'intermédiaire caché » (*man-in-the-middle attack*). Remarquez que, de toutes manières, vous n'avez probablement pas réellement besoin de rediriger les requêtes HTTPS vers Squid, dans la mesure où celui-ci ne mémorise pas les pages sécurisés.

2.4. Authentification auprès du mandataire

Il n'est pas possible de s'authentifier auprès d'un mandataire transparent. Voyez la FAQ Squid [http://www.squid-cache.org/Doc/FAQ/FAQ.html] pour (un peu) plus de détails.

3. Configurer le noyau

D'abord, il est nécessaire de s'assurer que notre noyau comporte les bonnes options de configuration. Si vous utilisez un noyau « prêt-à-porter » fourni par votre distribution, la gestion de mandataires transparents peut — ou non — être activée. Si vous n'en êtes pas sûr, le mieux à faire est de sauter cette section et, si les commandes de la prochaine section vous renvoient des erreurs bizarres, c'est probablement parce que votre noyau n'est pas correctement configuré.

Si votre noyau n'a pas été compilé avec les options de configuration permettant la gestion des mandataires transparents, vous devrez le recompiler. Recompiler un noyau est un processus complexe (du moins la première fois), et sort du sujet de ce document. Si vous avez besoin d'aide pour la compilation du noyau, reportez-vous au Guide pratique du noyau Linux [http://www.traduc.org/docs/HOWTO/lecture/Kernel-HOWTO.html]

Les options que vous devez sélectionner lors de la configuration du noyau sont les suivantes (remarque : si vous préférez des modules, certaines — mais pas toutes — peuvent être compilées comme modules. Heureusement, tout ce qui n'est pas modularisable peut être intégré à votre noyau) :

```
+ Dans « General setup »
        + « Networking support »
        + « Sysctl support »

+ Dans « Networking Options »
        + « Network packet filtering »
        + « TCP/IP networking »

+ Dans « Networking options » -> « IP : Netfilter configuration »
        + « Connection tracking »
        + « IP tables support »
        + « Full NAT »
        + « REDIRECT target support »

+ Dans « File system »
        + « /proc filesystem support »
```

Vous devez répondre *non* à « Fast switching » dans « Networking options ».

Une fois que vous aurez un nouveau noyau en état de fonctionner, vous pourrez avoir besoin d'activer le routage IP. Celui-ci permet à votre ordinateur de faire office de routeur. Dans la mesure où ce n'est pas ce que l'utilisateur moyen veut faire, cette option n'est pas activé par défaut et doit être activé de manière explicite au démarrage. Néanmoins, il est possible que votre distribution le fasse déjà pour vous. Pour le savoir, faites **`cat /proc/sys/net/ipv4/ip_forward`**. Si vous voyez « 1 » c'est bon. Sinon, faites **`echo '1' > /proc/sys/net/ipv4/ip_forward`**. Vous aurez certainement intérêt à ajouter cette commande dans le script de démarrage approprié (en fonction de votre distribution, il peut se trouver dans `/etc/rc.d`, `/etc/init.d`, ou carrément ailleurs).

4. Installer Squid

Il faut maintenant faire fonctionner Squid. Téléchargez l'archive la plus récente du code source depuis http://www.squid-cache.org. Assurez-vous que vous avez bien une version *stable* et non une version de développement. La version la plus récente à l'heure où j'écris ces lignes est squid-2.4.STABLE4.tar.gz. Remarquez qu'à ma connaissance vous devez utiliser squid-2.4 pour les noyaux Linux 2.4. La raison en est que le mécanisme par lequel le processus détermine l'adresse originale de destination a changé depuis Linux 2.2, et que le code nécessaire n'est présent qu'à partir de squid-2.4 (pour ceux que cela intéresse, auparavant l'appel `getsockname()` était bidouillé pour obtenir l'adresse originale de destination, alors que l'on utilise maintenant l'appel `getsockopt()` avec le niveau `SOL_IP` et l'option `SO_ORIGINAL_DST`).

Décompactez et extrayez l'archive (utilisez **`tar xzf`** ***`nom_du_fichier`***). Exécutez le script d'auto-configuration et dites-lui d'inclure le code destiné à Netfilter (**`./configure --enable-linux-netfilter`**), compilez (**`make`**) puis installez (**`make install`**).

Modifiez le fichier `squid.conf` (il devrait être sous `/usr/local/squid/etc/squid.conf`, à moins que vous n'ayez changé son chemin par défaut). Le fichier `squid.conf` est abondamment commenté. Il constitue pour certains sujets l'une des meilleures sources d'information sur Squid. Une fois que tout fonctionnera, je vous recommande de revenir en arrière et de le relire complètement. Mais pour l'instant, contentons-nous du minimum requis. Trouvez les directives suivantes, décommentez-les, et donnez-leur les valeurs appropriées :

```
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Ensuite, jetez un œil aux directives `cache_effective_user` et `cache_effective_group`. Si le couple utilisateur-groupe `nobody`/`nogroup` n'existe pas sur votre système (autant que je sache, ce n'est pas le cas dans de nombreuse distributions, y compris dans la RedHat 7.1), vous devrez le mettre en place, ou en créer un autre pour exécuter Squid. Je vous recommande chaudement de créer un couple utilisateur-groupe `squid`/`squid` pour faire tourner Squid, mais vous pouvez utiliser n'importe quel compte existant si vous le souhaitez.

Enfin, jetez un œil à la directive `http_access`. Sa valeur par défaut est habituellement `http_access deny all`. Ce qui empêche quiconque d'accéder à Squid. Provisoirement, vous pouvez changer cette valeur en `http_access allow all`, mais une fois le système opérationnel, il est fortement recommandé de lire la documentation consacrée aux listes de contrôle d'accès (ACL), et de configurer le mandataire de manière à ce que seules les personnes de votre réseau local (par exemple) puisse y accéder. Ceci peut paraître idiot, mais il est important de mettre en place de telles restrictions sur l'accès à votre cache. Les personnes bloquées par des pare-feu filtrants (tels que des filtres anti-pornographie, ou les filtres de pays totalitaires) font souvent de l'auto-stop sur les mandataires ouverts à tous vents et consomment votre bande passante.

Initialisez le répertoire utilisé pour la mémorisation des pages via la commande **`squid -z`** (vous devriez passer cette étape si Squid était déjà installé sur votre machine).

Maintenant, exécutez Squid en utilisant le script **`RunCache`** du répertoire `/usr/local/squid/bin/`. Si cela fonctionne, vous devriez être en mesure de régler les paramètres proxy de votre navigateur sur l'adresse IP de la machine où tourne Squid, et sur le port 3128 (à moins que vous n'ayez changé le numéro de port par défaut) et d'accéder à Squid comme à un mandataire normal.

Pour obtenir une aide complémentaire sur la configuration de Squid, reportez-vous à la FAQ de Squid sur http://www.squid-cache.org.

5. Installer iptables (Netfilter)

Iptables est une nouveauté des noyaux Linux 2.4, et remplace ipchains. Si votre distribution est fournie avec un noyau 2.4, iptables est probablement déjà installé. Dans le cas contraire vous devrez le télécharger (et probablement le compiler). Il est disponible sur http://netfilter.samba.org. Il est sans doute possible de trouver des paquets RPM binaires quelque part ailleurs, mais je n'ai pas cherché. Pour les curieux, le site de netfilter contient beaucoup de documentation.

Pour mettre en place les règles, vous devez connaître deux choses : l'interface par laquelle arrivent les requêtes des clients devant être transmises au serveur mandataire (j'utiliserai `eth0` dans l'exemple) et le port sur lequel Squid attend (à titre d'exemple, j'utiliserai la valeur par défaut `3128`).

Maintenant, voici les mots magiques de la mise en place d'un mandataire transparent :

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
         -j REDIRECT --to-port 3128
```

Vous devrez ajouter les commandes ci-dessus au script de démarrage approprié sous `/etc/rc.d`. Les lecteurs procédant à une mise à jour depuis un noyau 2.2 noteront que c'est la seule commande nécessaire. Les noyaux 2.2 exigeaient deux commandes supplémentaires pour empêcher les boucles de routage. L'infrastructure de netfilter est plus perfectionnée, et n'a besoin que de cette commande.

6. Mandataire transparent pour une machine distante

Maintenant, une question se pose naturellement, si l'on peut réaliser toutes ces astucieuses manœuvres pour rediriger les connexions HTTP vers des ports locaux, ne pourrait-on pas faire la même chose mais vers une machine distante (par exemple, dans le cas où la machine qui exécute Squid n'est pas la même que celle qui fait tourner **iptables**). La réponse est oui, mais cela demandes des mots magiques un peu différents. Si vous souhaitez uniquement une redirection vers la machine locale, ce qui est la cas habituel, vous pouvez ignorer ce chapitre.

Pour les besoins de l'exemple, supposons que nous ayons deux machines nommées *`machine-squid`* et *`machine-iptables`*, et qu'elles soient sur le réseau *`réseau-local`*. Dans les commandes ci-dessous, remplacez ces chaînes par les adresses ou les noms réels de vos réseau et machines.

Je présenterai ici deux approches différentes.

6.1. Première méthode (plus simple, mais non exhaustive)

- Sur laquelle tournera Squid, *`machine-squid`*, vous n'avez ni besoin d'iptables, ni d'indiquer au noyau une option spécifique. La seule chose nécessaire est Squid. Vous aurez en revanche *besoin*[1] d'indiquer à Squid l'option `http_accel` telle qu'elle est décrite ci-dessus.
- Sur la machine sur laquelle tournera iptables, *`machine-iptables`*, vous devrez configurer le

[1]Les versions précédentes de ce petit guide suggéraient que tel n'était pas le cas. C'était une erreur. Désolé d'avoir créé cette confusion.

noyau comme décrit dans Section 3, « Configurer le noyau » ci-dessus, à une exception près : vous n'avez pas besoin de l'option « REDIRECT target support ». Pour ce qui est des commandes iptables, vous aurez besoin de trois d'entre elles :

```
iptables -t nat -A PREROUTING -i eth0 -s !machine-squid \
         -p tcp --dport 80 -j DNAT --to machine-squid:3128

iptables -t nat -A POSTROUTING -o eth0 -s réseau-local \
         -d machine-squid -j SNAT --to machine-iptables

iptables -A FORWARD -i eth0 -o eth0 -s réseau-local \
         -d machine-squid -p tcp --dport 3128 -j ACCEPT
```

La première envoie les paquets de *`machine-iptables`* vers *`machine-squid`*. La seconde s'assure que la réponse soit renvoyée via *`machine-iptables`*, plutôt que directement au client (c'est très important !). La dernière s'assure que *`machine-iptables`* redirigera les paquets appropriés vers *`machine-squid`*. Il est possible qu'elle ne soit pas nécessaire. À vous de voir. Remarquez que nous avons spécifié `-i eth0` puis `-o eth0`, ce qui veut dire que nous utilisons `eth0` comme interface d'entrée (`-i`) et de sortie (`-o`). Si vos paquets entrent et sortent par des interfaces différentes, vous devrez ajuster ces commandes en conséquence.

Ajoutez ces commandes aux scripts de démarrage appropriés sous `/etc/rc.d/`

(Merci à Giles Coochey d'avoir aidé à l'écriture de cette section).

6.2. Seconde méthode (plus compliquée mais plus générale)

Notre première tentative marche bien, mais a un petit inconvénient : elle ne permet pas de gérer correctement les connexions HTTP/1.0 sans en-tête `Host`. Les connexions partiellement ou complètement compatibles HTTP/1.1, elles, marchent bien. En général, cela ne pose pas de problème, car la majorité des navigateurs modernes envoient l'en-tête `Host`. Cela pose problème dans le cas de certains petits programmes ou appareils embarqués, car ceux-ci n'émettent que des requêtes HTTP/1.0 très simples. Pour être capable de gérer correctement ce cas de figure, il faut en faire un peu plus.

- Sur *`machine-iptables`*, il est nécessaire d'activer les options suivantes du noyau :

```
IP: advanced router
IP: policy routing
IP: use netfilter MARK value as routing key
IP: Netfilter Configuration -> Packet mangling
IP: Netfilter Configuration -> MARK target support
```

Vous aurez également besoin des utilitaires iproute2. Votre distribution les a probablement déjà installés mais, dans le cas contraire, jetez un coup d'œil à ftp://ftp.inr.ac.ru/ip-routing/

La configuration de la machine nécessitera les commandes suivantes :

```
iptables -t mangle -A PREROUTING -j ACCEPT -p tcp --dport 80 -s machine-squid
iptables -t mangle -A PREROUTING -j MARK --set-mark 3 -p tcp --dport 80
ip rule add fwmark 3 table 2
ip route add default via squid-box dev eth1 table 2
```

Notez que les numéros choisis pour la marque de pare-feu (`3`) et pour la table de routage (`2`) sont complètement arbitraires. Si vous utilisez déjà un routage dirigé (*policy routing*) ou un marquage de pare-feu pour d'autres besoins, assurez-vous que vous choisissiez ici des numéros non

utilisés.

- Passons à `machine-squid`. Utilisez la commande suivante, qui devrait vous sembler remarquablement similaire à une commande vue précédemment.

```
iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 80 -j REDIRECT --to-port
```

Comme précédemment, ajoutez toutes ces commandes aux scripts de démarrage appropriés.

Voici une explication succincte de la façon dont cette seconde méthode fonctionne : dans la première méthode, nous avons utilisé la traduction d'adresse pour diriger les paquets vers l'autre machine. Ce qui implique une modification des paquets. Cette altération est la cause des défaillances mentionnés plus haut pour certains types de clients. Dans la méthode deux, nous utilisons un truc magique appelé routage dirigé (*policy routing*). Il faut tout d'abord sélectionner les paquets que l'on veut. Pour ce faire, nous marquons (via la cible MARK) tous les paquets destinés au port `80`, excepté ceux provenant de `machine-squid` elle-même. Habituellement, lorsque le noyau doit décider du routage des paquets, il utilise la table de routage consultable via la commande **route**. Pour le routage des paquets marqués, il utilisera une table spéciale ne comportant qu'une seule entrée, une passerelle par défaut pointant vers `machine-squid`. Les paquets visés seront donc joyeusement envoyés vers leur destin, sans subir aucune modification. Ce qui permettra de gérer correctement toutes les connexions, y compris les connexions HTTP/1.0. (Merci à Michal Svoboda d'avoir suggéré cette section et aidé à sa rédaction).

6.3. Première méthode… Et dans le cas où la machine iptables a une adresse IP dynamique ?

Si `machine-iptables` a une adresse IP dynamique (par exemple dans le cas d'une connexion ppp téléphonique ou d'une adresse assignée par DHCP sur un modem-câble), vous devrez alors apporter une légère modification aux commandes ci-dessus. Remplacez la seconde commande par celle-ci :

```
iptable -t nat -A POSTROUTING -o eth0 -s réseau-local \
        -d machine-squid -j MASQUERADE
```

Cette modification évite d'avoir à spécifier l'adresse IP de `machine-iptables` dans la commande. Dans la mesure où celle-ci change souvent, vous devriez modifier la commande à chaque fois. Cette modification vous épargnera donc beaucoup de travail.

7. Mandataire transparent configuré sur un pont réseau

Attention, nous entrons ici dans un domaine vraiment ésotérique. Si vous en avez besoin, vous saurez de quoi il s'agit. Merci à Lewis Shobbrook `<lshobbrook CHEZ fasttrack POINT net POINT au>` pour sa contribution à cette section.

Si vous essayez de configurer en mandataire transparent une machine Linux utilisée comme pont réseau, vous aurez besoin d'ajouter une commande supplémentaire à ce que nous avons dans Section 5, « Installer iptables (Netfilter) ». Plus précisément, vous aurez besoin de permettre explicitement les connexions à la machine sur le port `3128` (ou tout autre port sur lequel Squid est à l'écoute). En effet, si vous ne le faites pas, la machine fera suivre ces connexions directement via l'autre interface, comme le ferait tout bon petit pont. Voici les mots magiques :

```
iptable -A INPUT -i interface -d adresse_IP_du_pont \
        -s réseau-local -p tcp --dport 3128 \
```

```
        -m state --state NEW,ESTABLISHED -j ACCEPT
```

Remplacez *`interface`* par l'interface correspondant à *`adresse_IP_du_pont`* (en général, il s'agit de `eth0` ou `eth1`). Les personnes utilisant un pont pour la première fois devraient également prendre note du fait qu'il leur est possible de répéter la même commande en remplaçant `3128` par `ssh` afin de pouvoir administrer leur pont à distance.

8. Assembler le tout

Si jusqu'à présent tout s'est bien déroulé, installez-vous sur une autre machine, réglez son adresse de passerelle sur l'adresse IP de la machine qui exécute iptables, et naviguez. Pour vous assurer que les requêtes sont bien redirigées au travers du mandataire plutôt que d'être envoyées directement au serveur d'origine, il vous suffit de consulter le fichier journal : `/usr/local/squid/logs/access.log`

9. En cas de problème

Un problème spécifique est suffisamment fréquent pour mériter d'être mentionné ici. Si vous obtenez l'erreur suivante :

```
/lib/modules/2.4.2-2/kernel/net/ipv4/netfilter/ip_tables.o
init_modules: Device or resource busy
Hints: insmod errors can be caused by incorrect module parameters;
including invalid IO or IRQ parameters.

perhaps iptables or your kernel needs to be upgraded...
```

C'est sans doute que vous utilisez la distribution Red Hat 7.x. Les responsables de Red Hat, dans leur grande sagesse, on décidé de charger le module `ipchains` par défaut au démarrage. Il devait s'agir de préserver la compatibilité descendante pour ceux qui ne se sont pas encore mis à iptables. Cependant, le problème est que ipchains et iptables sont mutuellement incompatibles. Comme ipchains a été secrètement chargé par Red Hat, vous ne pouvez pas utiliser les commandes iptables. Pour vérifier si tel est bien votre problème, utilisez la commande **`lsmod`**, et vérifiez si le module nommé `ipchains` est présent. Si vous le trouvez, c'est que c'est bien là que se situe votre problème. Une solution à court terme est d'exécuter la commande **`rmmod ipchains`** avant d'exécuter les commandes **`iptables`**. Pour désactiver le chargement automatique du module `ipchains` au démarrage, essayez la commande suivante : **`/sbin/chkconfig --level 2345 ipchains off`** (merci à Rasmus Glud de m'avoir indiqué cette commande).

10. Informations complémentaires

Si vous avez besoin d'assistance, je vous recommande de consulter la FAQ de Squid ou sa liste de diffusion, sur http://www.squid-cache.org. Vous pouvez également me contacter en anglais à `<drk CHEZ unxsoft POINT com>`, et j'essaierai de répondre à vos questions si le temps me le permet (des fois oui, des fois non). SVP, SVP, SVP, envoyez-moi le résultat de la commande **`iptables -t nat -L`** et les portions significatives des fichiers de configuration dans votre courrier, sinon, je ne serai probablement pas en mesure de vous être très utile. S'il vous plaît, assurez-vous d'avoir lu l'intégralité de ce petit guide avant de poser une question. Bien que ce document aie été traduit dans de nombreuses langues, je ne pourrai répondre qu'aux questions posées en anglais, et j'en suis désolé.

11. Adaptation française

11.1. Traduction

La traduction française de ce document a été réalisée par Geneviève Gracian `<ggracian CHEZ free POINT fr>`.

11.2. Relecture

La relecture de ce document a été réalisée par Jean-Philippe Guérard <jean TIRET philippe POINT guerard CHEZ corbeaunoir POINT org>.